Approximating Distance Measures for the Skyline

Nirman Kumar
Department of Computer Science, University of Memphis, TN, USA
nkumar8@memphis.edu

Benjamin Raichel
Department of Computer Science, University of Texas at Dallas, TX, USA
benjamin.raichel@utdallas.edu

Stavros Sintos
Department of Computer Science, Duke University, Durham, NC, USA
ssintos@cs.duke.edu

Gregory Van Buskirk
Department of Computer Science, University of Texas at Dallas, TX, USA
greg.vanbuskirk@utdallas.edu

Abstract

In multi-parameter decision making, data is usually modeled as a set of points whose dimension is the number of parameters, and the *skyline* or *Pareto* points represent the possible optimal solutions for various optimization problems. The structure and computation of such points have been well studied, particularly in the database community. As the skyline can be quite large in high dimensions, one often seeks a compact summary. In particular, for a given integer parameter k, a subset of k points is desired which best approximates the skyline under some measure. Various measures have been proposed, but they mostly treat the skyline as a discrete object. By viewing the skyline as a continuous geometric hull, we propose a new measure that evaluates the quality of a subset by the Hausdorff distance of its hull to the full hull. We argue that in many ways our measure more naturally captures what it means to approximate the skyline.

For our new geometric skyline approximation measure, we provide a plethora of results. Specifically, we provide (1) a near linear time exact algorithm in two dimensions, (2) APX-hardness results for dimensions three and higher, (3) approximation algorithms for related variants of our problem, and (4) a practical and efficient heuristic which uses our geometric insights into the problem, as well as various experimental results to show the efficacy of our approach.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Skyline, Pareto optimal, Approximation, Hardness, Multi-criteria decision making

Digital Object Identifier 10.4230/LIPIcs.ICDT.2019.10

Related Version The full version of the paper is available online at [22], `http://utdallas.edu/~benjamin.raichel/stair.pdf`.

Funding B. Raichel and G. Van Burskirk are partially supported by NSF CRII Award 1566137 and CAREER Award 1750780. S. Sintos is supported by NSF under grants CCF-15-13816, CCF-15-46392, and IIS-14-08846, by an ARO grant W911NF-15-1-0408, and by BSF Grant 2012/229 from the U.S.-Israel Binational Science Foundation.

1 Introduction

When deciding which entry to return from a database where entries have multiple parameters, one must consider various criteria all at once. For example, given a collection of possible hotels, one seeks a hotel which costs less, is nearest to the desired location, and has a high rating. Typically one cannot optimize all these criteria simultaneously, though one would

never make a selection for which there is an alternative that is better in every way (i.e., cheaper, closer, and higher rated). This naturally leads to the definition of so called staircase or skyline points,[1] for which there are no strictly better alternatives, and each of which is a possible optimal solution depending on how the criteria are weighted.

Formally, we model our input as a set of n points P in $\mathbb{R}^d$. Given two points $p, q \in \mathbb{R}^d$, we say p *dominates* q, written $p \preceq q$, if $p_i \leq q_i$ for all i, where p_i (resp. q_i) denotes the ith coordinate of p (resp. q). Then the *staircase points* of P, denoted $\mathsf{S}(P)$, is the subset of points $p \in P$ such that p is *not* dominated by any point in $P \setminus \{p\}$.

While the staircase points model the set of possible solutions for various multi-criteria objectives, in many situations one desires a small subset of these points which well represent the tradeoffs in the solution space. For example, one may wish to present the user with a short list of search results to choose from. Moreover, the staircase might be large, particularly for high dimensional data sets, thus making a compact description highly desirable from a computational perspective. Here we propose a new measure to evaluate how well a subset of points approximates the staircase. Our work differs from previous attempts to define such a measure since rather than viewing $\mathsf{S}(P)$ as a discrete point set, instead we consider it as defining a continuous hull. Namely, we define the *staircase hull* of P, denoted $\mathsf{SH}(P)$, as the set of all points in $\mathbb{R}^d$ that are dominated by some point in P. Then for some integer parameter k, we seek the subset of at most k points $Q \subseteq P$ such that the distance between $\mathsf{SH}(Q)$ and $\mathsf{SH}(P)$ is minimized. To measure the distance between the two hulls we use the standard geometric notion of Hausdorff distance, which roughly speaking for two sets is defined as the maximum of all the distances from a point in one set to the closest point in the other set. For our problem we prove that this is equivalent to minimizing the maximum distance from any point in P to $\mathsf{SH}(Q)$. We argue that our geometric perspective in many ways better captures what it means to approximate the staircase. Moreover, our measure has additional advantages such as being translation invariant. In addition to defining this new measure, we provide a number of algorithmic, hardness, approximation, and experimental results as detailed below.

Previous work. Skyline/staircase points have been extensively used and studied, particularly in the database community (see for example [8]). In particular, it is known that for a set P of n points in $\mathbb{R}^d$, $\mathsf{S}(P)$ can be computed in $O(n(\log n + \log^{d-2} n))$ worst case time [24] for any d or in $O(n \log^{d-3} n \log\log n)$ for $d \geq 4$ [17]. In [12] they show a randomized algorithm for computing the $\mathsf{S}(P)$ in $O(n \log^{d-3})$ time for $d \geq 4$. There have also been a number of previous works on approximating the best subset of k input points to represent the skyline under various measures. Lin *et al.* [25] considered selecting k points so as to dominate the maximum possible number of points from P. This is an instance of the standard maximum coverage problem, for which the greedy algorithm achieves a $1 - 1/e$ approximation (to the number of dominated points). Later, Tao *et al.* [34] considered representing the skyline by a k-center clustering of the points in $\mathsf{S}(P)$, that is, minimize the maximum distance of a point in $\mathsf{S}(P)$ to the chosen subset, for which there are several standard 2-approximation algorithms to the optimal clustering radius. Koltun and Papadimitriou [21] consider what they call approximately dominating representatives (ADR), where a subset $Q \subseteq P$ is an ε-ADR if for every $p \in P$ there is some $q \in Q$ such that $(1 - \varepsilon)q$ dominates p (their actual definition

[1] In the title/abstract we use the term "skyline", more common in database papers, however in the rest of the paper we favor the term "staircase", common in computational geometry. In economics the term "Pareto" is often used.

differs slightly as they define dominating with the reverse inequality). Rather than focusing on minimizing the error radius as in our case and in [34], instead [21] keeps ε fixed and attempts to minimize k. As this variant is an instance of the well known Set-Cover problem, the greedy algorithm achieves an $O(\log n)$ approximation. In all three papers [25, 34, 21], among other things the authors give an exact polynomial time solution in 2 dimensions, and argue the problem is hard for dimensions 3 and higher. Several other measures have also been considered. Søholm *et al.* [33] select points so as to cover the maximum total area possible, giving an algorithm for the 2D case and some experimental results. Note they define dominance with the reverse inequality and so area is with respect to the coordinate axes, while in our case it would require defining some bounding box for the data. Magnani *et al.* [26] define a measure which combines two quantities which they call significance and diversity (where the user must specify the relative importance of each), and provide a hardness proof and experimental results.

More generally, there have been many other previously defined notions of compact approximate representations of the data, perhaps most notably the notion of clustering [36]. The so called k-regret minimizing set is a recent method of finding representative data points, which is used to approximate top-k queries [28, 13, 3, 11, 6, 23]. In the geometry community the notion of coresets, which are small subsets of the input approximately preserving some specified geometric property, has been widely studied [2]. Related to our notion of approximating the staircase hull, the case of approximating the convex hull [7] and the conic hull [9] was previously considered. These hull approximations more generally relate to a number of different types of matrix factorizations used in areas such as machine learning, including non-negative matrix factorization, CUR decomposition, and finding the top-k singular vectors. Note that points on the convex and conic hull boundaries are in general determined by a combination of multiple hull vertices. For the staircase hull we argue that each boundary point in some sense is determined by a single hull vertex, leading to stronger results as the problem can be modeled with asymmetric k-center.

Our results. As discussed above, here we introduce a new measure for approximating the staircase. Specifically, we seek the subset of k points $Q \subseteq P$ such that the Hausdorff distance between $\mathsf{SH}(Q)$ and $\mathsf{SH}(P)$ is minimized, which we refer to as the ***k-staircase problem***. In addition to introducing this new measure we have the following results:

1. In Section 3 we show that the k-staircase problem is APX-hard for any dimension $d \geq 3$. In particular, we give a simple argument to show it is hard to approximate within $\sqrt{2+\sqrt{3}} \approx 1.932$ for $d \geq 4$, and for $d = 3$ a more involved proof shows it is hard to approximate within a factor of $\sqrt{\frac{2+\sqrt{3}}{2+\sqrt{2}}} \approx 1.0455$.
2. Despite being hard to approximate for $d \geq 3$, in Section 4 we show that in two dimensions it can be solved optimally in $O(n \log^3 n)$ time.
3. In Section 2 we argue the k-staircase problem can actually be seen as an instance of the well known asymmetric k-center problem. In Section 5 this is used to give approximation algorithms for the k-staircase problem. In particular, in Section 5.1 we propose an $O(\log^*(n))$ approximation running in $O(n^2(d+\log^2 n))$ time, and an $O(\log^*(k))$ approximation running in polynomial time.
4. In Section 5.2, we show that if one is allowed to pick $2k$ rather than k points, then by making use of Well Separated Pair Decompositions (WSPD), for any fixed dimension d we can get a faster $O(kn \operatorname{polylog} n)$ time algorithm which still achieves an $O(\log^*(n))$ approximation for the k-staircase problem. This is the only result where we assume d is fixed. (The algorithm works for any d, though the approximation and time degrade.)

5. In Section 6 we propose a new heuristic for the k-staircase problem which is based on finding what we call relaxed CCV[2] points. Such points have nice symmetric geometric properties, and we argue that any point set always contains at least one such point. While certain contrived worst case examples make it difficult to provide worst case guarantees, we show experimentally in Section 6.1 that approximating the staircase hull using relaxed CCV points consistently gives a smaller amount of error than the guaranteed $O(\log^*(n))$-approximation algorithm of [30]. Moreover, our heuristic is simple, easy to implement, and fast, thus making it quite practical. Finally, in Section 6.2, we compare our measure to several others by using NBA player statistics to predict NBA All Star teams. Not only does our measure always select the most all star players, but also it is inversely proportional to the number of selected players, while for other measures the correspondence is less consistent.

Due to space limitations, some of the proofs can be found in the full version of the paper [22].

The Hausdorff distance uses the standard L_2 norm for distances between points, however, it can be generalized to any L_p norm.[3] Intuitively, the L_1 norm focuses on the average value over the dimensions, while L_∞ focuses on the dimension which is largest. We argue our algorithmic results hold for any L_p norm, thus allowing the user to tune the dial between these two extremes in order to best suit a given application.

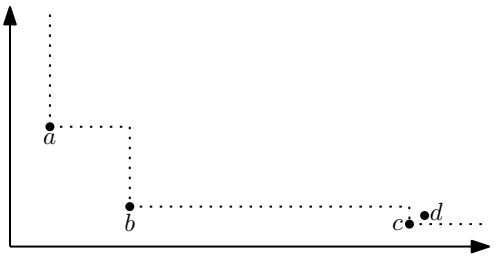

Figure 1.1 For $k = 2$, our solution is a and b, k-center a or b and c or d, max cover a or b and c.

Understanding and comparing measures. As mentioned above, one of the main ways in which our measure differs from [25] and [34] is that it considers the staircase as defining a continuous hull rather than a discrete point set. In particular, consider the simple four point example shown in Figure 1.1. For $k = 2$, in the maximum coverage measure of [25] the optimal solution contains either a or b along with the point c. For the k-center clustering measure of [34] the optimal solution contains exactly one of a or b and exactly one of c or d. On the other hand, for our measure the optimal solution contains points a and b. Arguably a and b represent real tradeoffs between the horizontal and vertical coordinate values, whereas c and d have a negligible advantage in the vertical coordinate over b but are far worse in the horizontal coordinate. In other words, when considering the discrete points, c and d are far from a and b. However, when considering the continuous hulls, c and d are very close to the hull $\mathsf{SH}(\{a, b\})$, yet b has a larger distance to the hull $\mathsf{SH}(\{a, c\})$.

There are several additional nice properties of our measure. First, it is natural from a geometric perspective as it is just the standard geometric notion of Hausdorff distance between two objects. Second, it is translation invariant, i.e., the optimal solution does not

[2] The notion of CCV points, which stands for "center capturing vertex", was introduced in [30] to solve the asymmetric k-center problem approximately.

[3] Here we use the standard L_p notation, though later we use L_α as p is often used to denote a point.

change if all points are translated by the same amount (where the amount in each coordinate can differ). This is also true of the measures in [25] and [34], however, it is not the case for [33] and [21], making it difficult to compare their measure to ours as theirs strongly depends on the choice of origin. A number of other properties, such as scale invariance or stability, have been used to characterize previous approaches for approximating the staircase. In the full version of the paper [22] we provide a table and description comparing our measure with others, showing the tradeoffs of these various properties. Rather than focusing on any one of these properties, here we advocate the benefit of taking into account the continuous hull, and give some partial justification of this approach with our experimental results in Section 6.

In order to maximize the benefit of our measure, there are several preprocessing steps that one may wish to take. First observe that depending on what the coordinate values represent, in some coordinates one might desire a larger value and in others a smaller one (e.g. hotel rating vs. price). To handle this one can negate all values in coordinates for which large values are better, and then translation invariance implies we can shift the points to remove negative values (if so desired). Translation invariance also implies one can translate the point set such that in each coordinate the minimum value of any data point is zero. Also, note that if one scales the data by the same factor in every coordinate then the optimal solution does not change, hence one can scale so the data points lie in the unit hypercube if so desired. On the other hand our measure is sensitive to scaling by different values in each coordinate, and this fact can be used to encode our preferences over the coordinates. Specifically, if one scales such that the maximum value is exactly 1 in each coordinate then this attempts to put each coordinate on equal footing. Alternatively, if one knows certain attributes are more important than others, one can weight the scaling differently in each coordinate to take this into account. Note that all this can easily be done in linear time.

2 Problem Statement and Connection to Asymmetric K-center

2.1 Definitions and problem statement

Given points $p, q \in \mathbb{R}^d$, we say p *dominates* q, written $p \preceq q$, if $p_i \leq q_i$ for all i, where p_i (resp. q_i) denotes the ith coordinate of p (resp. q). For a point set $P \subset \mathbb{R}^d$, let $\mathsf{S}(P)$ denote the set of *staircase points* of P, which is the set of points $p \in P$ such that p is *not* dominated by any point in $P \setminus \{p\}$. Finally, let $\mathsf{SH}(P)$ denote the *staircase hull* of P, which is the subset of points in $\mathbb{R}^d$ dominated by some point in P. Notice that $\mathsf{S}(P) \subset P \subset \mathsf{SH}(P)$ and $\mathsf{SH}(P) = \mathsf{SH}(\mathsf{S}(P))$ as dominance is transitive.

Let $\mathsf{d}_\alpha(x, y) = ||x - y||_\alpha$ denote the distance between $x, y \in \mathbb{R}^d$ with respect to the L_α norm[4], where $\alpha \geq 1$. For (closed) sets $X, Y \subset \mathbb{R}^d$, let $\mathsf{d}_\alpha(X, Y) = \min_{x \in X, y \in Y} ||x - y||_\alpha$ be the distance between point sets X and Y with respect to the L_α norm. If X is a singleton $\{x\}$, let $\mathsf{d}_\alpha(x, Y)$ denote $\mathsf{d}_\alpha(X, Y)$. For point sets X and Y, let $\mathsf{d}_{\mathsf{H}_\alpha}(X, Y) = \max\{\max_{x \in X} \mathsf{d}_\alpha(x, Y), \max_{y \in Y} \mathsf{d}_\alpha(y, X)\}$ denote the Hausdorff distance with respect to the L_α norm, between X and Y. Typically, we consider the case where $Y \subseteq X$; here $\mathsf{d}_{\mathsf{H}_\alpha}(X, Y) = \max_{x \in X} \mathsf{d}_\alpha(x, Y)$.

▶ **Problem 1** (k-staircase problem). *Given a set of n points $P \subset \mathbb{R}^d$ and a parameter k, find a subset $Q \subseteq P$ with $|Q| \leq k$ which minimizes $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q))$.*

[4] The L_α norm is defined for $\alpha \geq 1$, as $||u||_\alpha = \left(\sum_{j=1}^{d} |u_j|^\alpha\right)^{1/\alpha}$.

For $Q \subseteq P$, the value $r = \mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q))$ is *the radius of* Q. Throughout the paper, we say that Q *r-covers* P when $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q)) \leq r$. For an optimal solution Q^* to Problem 1, we use r^* to denote its corresponding optimal radius. A subset $Q \subseteq P$ of size k is called a c-approximation for Problem 1 if the radius of Q is at most cr^*.

2.2 Properties and asymmetric k-center

Above we defined the notions of dominance, staircase points, and the staircase hull. In this section we provide further definitions and properties, to ultimately reduce Problem 1 to the well known asymmetric k-center problem, which we now define. Then, using the $O(\log^* n)$-approximation algorithm for the asymmetric k-center problem [30], in Section 5 we can show how to approximate the k-staircase problem.

▶ **Problem 2** (Asymmetric k-center)**.** *Given a set of n points P, with a corresponding asymmetric distance function $f : P \times P \to \mathbb{R}^{\geq 0}$, and a parameter k, find a subset $Q \subseteq P$ of at most k points which minimizes* $\max_{p \in P} f(Q, p) = \max_{p \in P} \min_{q \in Q} f(q, p)$.

In the above definition, by an *asymmetric* distance function we mean a function $f : P \times P \to \mathbb{R}^{\geq 0}$ which satisfies the *directed triangle inequality*: for any three points $a, b, c \in P$ we have $f(a, c) \leq f(a, b) + f(b, c)$. However, f may not be symmetric, i.e., in general $f(a, b) \neq f(b, a)$, and thus it is not a metric.

To reduce Problem 1 to Problem 2, we need to provide a number of definitions and structural properties. First, observe that the staircase hull is the union of a set of regions determined by individual points of P (unlike the convex hull for example). Using $\mathsf{SH}(p)$ to denote the subset of points in $\mathbb{R}^d$ dominated by p, we observe that $\mathsf{SH}(P) = \bigcup_{p \in P} \mathsf{SH}(p)$.

Next, we introduce some useful notation that we use throughout the paper. For a vector $u = (u_1, \ldots, u_d) \in \mathbb{R}^d$ define $||u||_\alpha^- = \left(\sum_{u_j < 0} |u_j|^\alpha\right)^{1/\alpha}$, and $||u||_\alpha^+ = \left(\sum_{u_j \geq 0} u_j^\alpha\right)^{1/\alpha}$. Observe that, $||u||_\alpha^- = ||-u||_\alpha^+$ and $||u||_\alpha^+ = ||-u||_\alpha^-$.

We first provide a simple expression for the distance from a point $p = (p_1, \ldots, p_d) \in \mathbb{R}^d$ to the region $\mathsf{SH}(q)$ of another point $q = (q_1, \ldots, q_d)$.

▶ **Lemma 3.** *For any points $p, q \in \mathbb{R}^d$, we have that,* $\mathsf{d}_\alpha(p, \mathsf{SH}(q)) = ||q - p||_\alpha^+$.

Proof. The region $\mathsf{SH}(q)$ is the set of points $(x_1, \ldots, x_d) \in \mathbb{R}^d$ such that $x_j \geq q_j$ for $j = 1, \ldots, d$. The L_α distance from p to x to the power of α is $(||x - p||_\alpha)^\alpha = \sum_{j=1}^d |x_j - p_j|^\alpha$. It is easy to see that to minimize this expression over $\mathsf{SH}(q)$ one should take the point $(x_1, \ldots, x_d)$ which has $x_j = p_j$ for all j such that $p_j > q_j$ and one should let $x_j = q_j$ otherwise. The result follows. ◀

The above implies that for any two points p and q, one can define a natural notion of *directed distance* (or equivalently asymmetric distance) from p to q, namely $\vec{\mathsf{d}}_\alpha(p, q) = \mathsf{d}_\alpha(q, \mathsf{SH}(p))$. Note the change in the order of the arguments. Alternatively one could define the directed distance from p to q as $\mathsf{d}_\alpha(p, \mathsf{SH}(q))$, which is natural when viewing p as traveling to $\mathsf{SH}(q)$. However, our definition is more natural when considering how well $\mathsf{SH}(p)$ covers q, ultimately helping relate our problem to the asymmetric k-center problem.

By Lemma 3, we have that,

$$\vec{\mathsf{d}}_\alpha(p, q) = \mathsf{d}_\alpha(q, \mathsf{SH}(p)) = ||p - q||_\alpha^+ = ||q - p||_\alpha^- = \left(\sum_{j : q_j < p_j} |q_j - p_j|^\alpha \right)^{1/\alpha}.$$

We say that p r-covers q when $\vec{\mathsf{d}}_\alpha(p, q) \leq r$.

A fundamental inequality obeyed by our directed distances is the following:

▶ **Lemma 4.** *Let x be a point dominated by y, i.e., $x \in \mathsf{SH}(y)$. Then, for any point p we have that, $\vec{\mathsf{d}}_\alpha(p,x) \leq \vec{\mathsf{d}}_\alpha(p,y)$, and $\vec{\mathsf{d}}_\alpha(y,p) \leq \vec{\mathsf{d}}_\alpha(x,p)$.*

Proof. We only prove the first inequality, as the proof of the second is similar. We have that $(||x-p||_\alpha^-)^\alpha = \sum_{j:x_j<p_j} |x_j-p_j|^\alpha$. Since $x \in \mathsf{SH}(y)$ we have that for all $j=1,\ldots,d$, $y_j \leq x_j$. For any j where $x_j < p_j$ clearly $y_j < p_j$ as well, and moreover, $|y_j-p_j|^\alpha \geq |x_j-p_j|^\alpha$. Thus, $(||x-p||_\alpha^-)^\alpha \leq (||y-p||_\alpha^-)^\alpha$. It follows that, $(\vec{\mathsf{d}}_\alpha(p,x))^\alpha = (||x-p||_\alpha^-)^\alpha \leq (||y-p||_\alpha^-)^\alpha = (\vec{\mathsf{d}}_\alpha(p,y))^\alpha \Rightarrow \vec{\mathsf{d}}_\alpha(p,x) \leq \vec{\mathsf{d}}_\alpha(p,y)$. ◀

The above lemma can now be used to show that our directed distances satisfy the directed triangle inequality, a fact not immediately apparent from our distance formula (Lemma 3), and which we require later in order to apply previous results for asymmetric k-center.

▶ **Lemma 5.** *The directed distance function $\vec{\mathsf{d}}_\alpha(\cdot,\cdot)$ obeys the directed triangle inequality. That is, for any three points $a,b,c \in \mathbb{R}^d$ we have $\vec{\mathsf{d}}_\alpha(a,c) \leq \vec{\mathsf{d}}_\alpha(a,b) + \vec{\mathsf{d}}_\alpha(b,c)$.*

Proof. Let z be the closest point in $\mathsf{SH}(b)$ to c, i.e., $z = \operatorname{argmin}_{x\in\mathsf{SH}(b)} \mathsf{d}_\alpha(c,x)$, and let z' be the closest point in $\mathsf{SH}(a)$ to z, i.e., $z' = \operatorname{argmin}_{x\in\mathsf{SH}(a)} \mathsf{d}_\alpha(z,x)$. Then,

$$\vec{\mathsf{d}}_\alpha(a,c) \leq \mathsf{d}_\alpha(c,z') \leq \mathsf{d}_\alpha(c,z) + \mathsf{d}_\alpha(z,z') = \vec{\mathsf{d}}_\alpha(b,c) + \vec{\mathsf{d}}_\alpha(a,z) \leq \vec{\mathsf{d}}_\alpha(b,c) + \vec{\mathsf{d}}_\alpha(a,b).$$

The first inequality holds because $z' \in \mathsf{SH}(a)$ and $\vec{\mathsf{d}}_\alpha(a,c) = \mathsf{d}_\alpha(c,\mathsf{SH}(a))$. The second inequality holds because the L_α norm satisfies the triangle inequality. The third equality follows from the definition of z and z'. Since z is dominated by b, by Lemma 4 we have that $\vec{\mathsf{d}}_\alpha(a,z) \leq \vec{\mathsf{d}}_\alpha(a,b)$ so the result follows. ◀

We now argue that for any subset $Q \subseteq P$ in Problem 1, minimizing $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P),\mathsf{SH}(Q))$ can instead be thought of as minimizing the distance from the finite point set P to the hull $\mathsf{SH}(Q)$. Thus as $\mathsf{SH}(Q) = \bigcup_{q\in Q} \mathsf{SH}(q)$, this in turn allows us to think of the problem as selecting Q so as to minimize directed distances from P to Q, i.e., a clustering problem.

For a point set Q and a point x, in the following we use the notation $\vec{\mathsf{d}}_\alpha(Q,x) = \min_{q\in Q} \vec{\mathsf{d}}_\alpha(q,x) = \min_{q\in Q} \mathsf{d}_\alpha(x,\mathsf{SH}(q))$. Observe that $\min_{q\in Q} \mathsf{d}_\alpha(x,\mathsf{SH}(q)) = \mathsf{d}_\alpha(x,\mathsf{SH}(Q))$, and hence $\vec{\mathsf{d}}_\alpha(Q,x) = \mathsf{d}_\alpha(x,\mathsf{SH}(Q))$.

▶ **Lemma 6.** *For $Q \subseteq P$, $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P),\mathsf{SH}(Q)) = \max_{p\in P} \mathsf{d}_\alpha(p,\mathsf{SH}(Q)) = \max_{p\in\mathsf{S}(P)} \mathsf{d}_\alpha(p,\mathsf{SH}(Q))$.*

Proof. Note that $\mathsf{S}(Q) \subseteq \mathsf{S}(P)$ since $Q \subseteq P$, and so

$$\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P),\mathsf{SH}(Q)) = \max_{p\in\mathsf{SH}(P)} \mathsf{d}_\alpha(p,\mathsf{SH}(Q)).$$

As $\mathsf{S}(P) \subseteq P \subseteq \mathsf{SH}(P)$ we therefore have

$$\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P),\mathsf{SH}(Q)) \geq \max_{p\in P} \mathsf{d}_\alpha(p,\mathsf{SH}(Q)) \geq \max_{p\in\mathsf{S}(P)} \mathsf{d}_\alpha(p,\mathsf{SH}(Q)).$$

For the other direction, for any point $p \in P$, let $f(p) \in \mathsf{S}(P)$ be any point that dominates p. For any point $q \in Q$, it follows by Lemma 4 that $\vec{\mathsf{d}}_\alpha(q,p) \leq \vec{\mathsf{d}}_\alpha(q,f(p))$ and as such taking the minimum over Q we get that $\vec{\mathsf{d}}_\alpha(Q,p) \leq \vec{\mathsf{d}}_\alpha(Q,f(p))$. Now taking the maximum over $p \in P$ we get that

$$\max_{p\in P} \mathsf{d}_\alpha(p,\mathsf{SH}(Q)) = \max_{p\in P} \vec{\mathsf{d}}_\alpha(Q,p) \leq \max_{p\in P} \vec{\mathsf{d}}_\alpha(Q,f(p)) \leq \max_{p\in\mathsf{S}(P)} \vec{\mathsf{d}}_\alpha(Q,p) = \max_{p\in\mathsf{S}(P)} \mathsf{d}_\alpha(p,\mathsf{SH}(Q)).$$

Similarly, because for each point $x \in \mathsf{SH}(P)$ there is some point $p \in P$ dominating x we can argue that $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P),\mathsf{SH}(Q)) \leq \max_{p\in P} \mathsf{d}_\alpha(p,\mathsf{SH}(Q))$. ◀

Note we can assume that any optimal solution Q to Problem 1 lies on the staircase $\mathsf{S}(P)$, since otherwise we can choose a dominating point on $\mathsf{S}(P)$ for each point of Q, defining a new set Q' such that $\mathsf{SH}(Q') \supseteq \mathsf{SH}(Q)$ and so $\max_{p\in\mathsf{S}(P)} \mathsf{d}_\alpha(p, \mathsf{SH}(Q')) \leq \max_{p\in\mathsf{S}(P)} \mathsf{d}_\alpha(p, \mathsf{SH}(Q))$. Thus the above implies that in Problem 1 we can consider the original input P as consisting entirely of staircase points (i.e., $\mathsf{S}(P) = P$), since otherwise as an initial step we can throw out all non-staircase points. Moreover, since $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q)) = \max_{p\in P} \mathsf{d}_\alpha(p, \mathsf{SH}(Q))$ $= \max_{p\in P} \vec{\mathsf{d}}_\alpha(Q, p)$, the above implies Problem 1 can easily be seen as an instance of asymmetric k-center, for our asymmetric distance function $\vec{\mathsf{d}}_\alpha(\cdot,\cdot)$.

3 Hardness

In this section we mention APX-hardness results for the k-staircase problem for $d \geq 3$ using the L_2 norm. For the L_∞ norm, the hardness proof by Koltun and Papadimitriou [21] can be used to show that the problem is NP-hard, but there is no known APX-hardness proof.

Our main result for all $d \geq 4$ is,

▶ **Theorem 7.** *The k-staircase problem for the L_2 norm is* NP*-hard to approximate in $\mathbb{R}^d$, for any $d \geq 4$, within a factor better than $\sqrt{2+\sqrt{3}} \approx 1.932$.*

Proof. Since the results hold only for the L_2 norm, we skip the subscript 2 from the statements. We show the following claim which is required to establish the gap-preserving nature of a reduction from the k-center problem in $\mathbb{R}^2$ to the k-staircase problem in $\mathbb{R}^4$. We prove that there is a mapping $\phi : \mathbb{R}^2 \to \mathbb{R}^4$, such that for any two points $p, q \in \mathbb{R}^2$ we have that, $\|p - q\| = \vec{\mathsf{d}}(\phi(p), \phi(q)) = \vec{\mathsf{d}}(\phi(q), \phi(p))$.

The mapping ϕ is defined by $\phi((x, y)) = (x, -x, y, -y)$. Given $p = (p_x, p_y)$ and $q = (q_x, q_y)$, we have that $\|p - q\| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. Now, $\phi(p) - \phi(q) = (p_x - q_x, q_x - p_x, p_y - q_y, q_y - p_y)$. If $p_x \neq q_x$, exactly one of $p_x - q_x$ and $q_x - p_x$ is greater than 0. A similar statement is true for $p_y - q_y, q_y - p_y$. Thus, $||\phi(p) - \phi(q)||^+ = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. If $p_x = q_x$, then neither of $(p_x - q_x)^2, (q_x - p_x)^2$ appear in the expression for $||\phi(p) - \phi(q)||^+$ but there is no contribution to $\|p - q\|$ from the x-coordinate as well. A similar statement is true for the y-coordinate. Thus, $\vec{\mathsf{d}}(\phi(p), \phi(q)) = ||\phi(p) - \phi(q)||^+ = \|p - q\|$. An analogous argument shows that $\|p - q\| = ||\phi(p) - \phi(q)||^- = ||\phi(q) - \phi(p)||^+ = \vec{\mathsf{d}}(\phi(q), \phi(p))$ as well. The claim follows.

The k-center problem in $\mathbb{R}^2$ is polynomial time reducible to the k-staircase problem in $\mathbb{R}^4$. Indeed, for a set P of n points in $\mathbb{R}^2$ we consider the set $P' = \{\phi(p) : p \in P\} \subset \mathbb{R}^4$. Clearly P' can be constructed in $O(n)$ time. Moreover, our claim implies that for any (at most) k points $p_1, \ldots, p_k \in P$ which give a clustering radius R, the corresponding points $\phi(p_1), \ldots, \phi(p_k)$ will give a value of R for the clustering radius in the k-staircase problem for P'. (Interestingly note that in fact all the mapped points lie on the staircase.) The minimum clustering radii thus are equal. Any APX-hardness result for the k-center problem in $\mathbb{R}^2$ can be combined with this polynomial time reduction to give the same hardness for the k-staircase problem in $\mathbb{R}^4$. Feder and Greene [16] proved that the k-center problem in $\mathbb{R}^2$ cannot be approximated to a factor better than $\frac{(1+\sqrt{7})}{2} \approx 1.823$. There is also such a result by Mentzer [27], although it is not that well known. Mentzer shows that the k-center problem in $\mathbb{R}^2$ is hard to approximate within a factor of $\sqrt{2+\sqrt{3}} \approx 1.932$. Using Mentzer's result [27] we conclude the result of the theorem. ◀

For $d = 3$, a non-trivial approximate embedding argument and Mentzer's result [27] implies,

▶ **Theorem 8.** *The k-staircase problem for the L_2 norm is* NP*-hard to approximate in $\mathbb{R}^3$ within a factor better than $\frac{\sqrt{2+\sqrt{3}}}{\sqrt{2+\sqrt{2}}} \approx 1.0455$.*

The proof of the last theorem can be found in [22].

Finally, we would like to point out the paper by Chuzhoy et al. [14], who showed that the asymmetric k-center problem is hard to approximate up to a factor $\log^* n - \Theta(1)$ unless NP $\subseteq$ DTIME$(n^{\log\log n})$. An interesting open question is if their reduction can be used to show that our k-staircase problem is hard to approximate by a factor better than $\log^* n$. The authors construct a complicated graph for their reduction. In order to use their construction we would have to embed their graph to the Euclidean space, such that i) the nodes are represented by points on the staircase, and ii) capture their (directed) graph distances with our geometric directed distances. Because of the complexity of their graph we could not find such a geometric embedding, and hence we leave it as an open problem. For details see the full version of the paper [22].

4 Exact Algorithm in 2D

In this section we present a polynomial time exact algorithm for the k-staircase problem in 2 dimensions for any L_α norm with $\alpha \geq 1$.

▶ **Theorem 9.** *Given a set of n points $P \subset \mathbb{R}^2$, an integer parameter $k \leq n$, and a real parameter $\alpha \geq 1$, a set $Q \subseteq P$ of size at most k can be computed in $O(n \log^3 n)$ time, such that $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q))$ is minimized.*

Intuitively, the staircase problem is easier in 2D because of the ordering of the points along the x and the y axis: If the x-coordinate of $p \in \mathsf{S}(P)$ is smaller than the x-coordinate of $q \in \mathsf{S}(P)$, then the y-coordinate of p is larger than the y-coordinate of q. Let p_1, p_2 denote the x, y-coordinates of a point p and let $p, q \in \mathsf{S}(P)$ be two points in the staircase such that $p_1 \geq q_1$ and $p_2 \leq q_2$. Then, for any $\alpha, \beta \geq 1$ we have that $\vec{\mathsf{d}}_\alpha(p,q) = ||p-q||_\alpha^+ = \left((p_1 - q_1)^\alpha\right)^{1/\alpha} = p_1 - q_1 = \left((p_1 - q_1)^\beta\right)^{1/\beta} = \vec{\mathsf{d}}_\beta(p,q)$. Thus, all the norms are equivalent, and so without loss of generality, in what follows, $\alpha = 1$. We now describe our algorithm. Assume that we have a procedure Cover(P, r), which given a radius $r \geq 0$ returns a set $Q_r \subseteq \mathsf{S}(P)$ of minimum size satisfying $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q_r)) \leq r$. Notice that by definition r^* is the minimum value of r such that $|Q_r| \leq k$. Since r^* is one of the $\Theta(n^2)$ distances $\vec{\mathsf{d}}(p_i, p_j)$, we could run a binary search over all possible distances to find r^*. For a given distance r we would need to decide if $r < r^*$ or $r \geq r^*$. To do this, we use Cover(P, r) and check if the set Q_r it returns has $|Q_r| \leq k$. A naive implementation of this algorithm would enumerate all the $\Theta(n^2)$ distances, sort them, and do the binary search over them, but this would clearly take at least quadratic time. We now show how to implement the algorithm in near-linear time.

Imagine an array containing all the $\Theta(n^2)$ distances in sorted order. As remarked above, we cannot directly compute this array in near-linear time. However we do not need access to the entire array. A typical query during binary search is : "return the m-th element of this array". Assume that finding the m-th smallest distance among the points in $\mathsf{S}(P)$, and the procedure Cover$(\cdot)$, both take $O(n \operatorname{polylog}(n))$ time. Then, all the $O(\log n)$ distance queries for the binary search and all the calls to Cover$(\cdot)$ would take $O(n \operatorname{polylog}(n))$ time. Following this scheme, we now present an algorithm with overall $O(n \log^3 n)$ running time.

Algorithm 1: Cover(P, r).

Input : P, r
Output : $Q_r \subseteq P$ with $|Q_r| \leq k_r$ and $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q_r)) \leq r$

```
V = ∅
for p ∈ S(P) do
    V_p = {q | d⃗(p, q) ≤ r,  q ∈ S(P)}
    V = V ∪ {V_p}
F = Min_Set_Cover(S(P), V)
Q = {p | V_p ∈ F}
Return Q
```

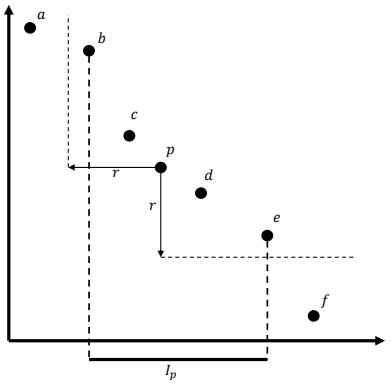

Figure 4.1 The interval I_p is defined by the x-coordinate of point b and of point e.

The procedure Cover(P, r). Let k_r be the smallest size of a set of centers Q_r such that $\mathsf{d}_{\mathsf{H}_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q_r)) \leq r$. The pseudocode of Cover(P, r) can be seen in Algorithm 1.

Correctness. For each point $p \in \mathsf{S}(P)$, the set V_p contains the points that are covered by p within radius r. Then we solve the min-set-cover instance where the elements are the points in $\mathsf{S}(P)$ and V is the family of sets. If the solution to the min-set-cover is optimal, the correctness of the algorithm follows.

Even though min-set-cover is an NP-hard problem, some special instances can be solved optimally in polynomial time. We show that the set system $(\mathsf{S}(P), V)$ can be mapped to a set system where the elements are points on a line and the sets are intervals intersected with the point set. This special case of the min-set-cover problem is the well known unweighted interval point cover problem (IPCP) and for n intervals and n points it can be solved optimally in $O(n \log n)$ time [4, 35, 20]. The mapping of points is simple. For a point $p = (p_1, p_2)$ it is just the projection of p to p_1 on the x-axis. Let P_1 denote this projected set of points. The following lemma shows that the set of points q for which $\vec{\mathsf{d}}_\alpha(p, q) \leq r$ corresponds to an interval on the x-axis.

▶ **Lemma 10.** *Let $p, q, z \in \mathsf{S}(P)$. If q_1 is between p_1 and z_1 and $\vec{\mathsf{d}}_\alpha(p, z) \leq r$ then $\vec{\mathsf{d}}_\alpha(p, q) \leq r$.*

Proof. We show it for the case where $p_1 \leq q_1 \leq z_1$, since the case of $z_1 \leq q_1 \leq p_1$ is similar. Since $p, q, z \in \mathsf{S}(P)$ and $p_1 \leq q_1 \leq z_1$, it also holds that $p_2 \geq q_2 \geq z_2$. We have that $\vec{\mathsf{d}}_\alpha(p, z) = p_2 - z_2 \leq r$. We conclude that $\vec{\mathsf{d}}_\alpha(p, q) = p_2 - q_2 \leq p_2 - z_2 \leq r$. ◀

The interval I_p for a point p can be computed as follows. The left end-point is just $p_1 - r$ but we can clip it to a point in P_1 by doing a successor search for $p_1 - r$ among the set of points P_1. The right end-point corresponds to the x-coordinate q_1 of a point $q = (q_1, q_2)$ such that $q_1 \geq p_1$ and is the largest value such that $p_2 - q_2 \leq r$. To compute this we can do a predecessor search for $p_2 + r$ in the projection of points P_2 on the y-axis. Once we find this point q_2, the left end-point of I_p is determined by q_1. In Figure 4.1, we show an example of the interval I_p for a point p. Using the above construction and Lemma 10 we can argue that each set V_p corresponds to an interval I_p on the x-axis and hence the set cover instance defined by $(\mathsf{S}(P), V)$ is an instance of the IPCP.

Complexity of Cover(P, r). We can compute the staircase $\mathsf{S}(P)$ of P in $O(n \log n)$ time. The (sorted) projected point sets P_1, P_2 as above can be constructed in additional $O(n)$ time. For each point p the set V_p is specified by the interval I_p which needs two predecessor

searches. Thus each such interval can be computed in $O(\log n)$ time. The min-set-cover for the IPCP instance can be computed in $O(n \log n)$ time. Thus, overall Cover(P, r) takes $O(n \log n)$ time.

Search over the possible radii. Let $Z = n(n-1)$ denote the total number of distances $\vec{\mathsf{d}}_\alpha(p, q)$ between points $p, q \in P$. As remarked above, given m with $1 \leq m \leq Z$, we need a method to compute the m-th smallest distance in this set. We now show how to do this in near-linear time, more specifically, $O(n \log^2 n)$ time. We first discuss another problem which will be used. Let A, B be two sorted arrays of real numbers with length M each. A classic exercise is to compute the m-th smallest element in the union of the two arrays in sublinear time and it is well known that it can be computed in $O(\log M)$ time. The C implementation of such an algorithm can be found in [22]. This algorithm makes at most $O(\log M)$ comparisons between the elements of A, B. The set R of all the Z distances $\vec{\mathsf{d}}_\alpha(p, q)$ can be written as the union of two sets A_x and A_y defined as, $A_x = \{p_1 - q_1 \mid p_1 \geq q_1\}$, and $A_y = \{p_2 - q_2 \mid p_2 \geq q_2\}$. Notice that $|A_x| = |A_y| = Z/2$, so let $M = Z/2$. So, our goal is to compute the m-th smallest element of $A_x \cup A_y$. To use the algorithm just mentioned, we need access to the i-th smallest element of A_x (or a similar query for A_y). Fortunately, we do not need to explicitly compute A_x, A_y. In [31, 32] the authors show that the i-th smallest L_1 distance of n points on a line can be computed in $O(n \log n)$ time. The distances in A_x, A_y can be seen as the L_1 distances among the projections of $\mathsf{S}(P)$ on the two axes, namely, among the points of P_1 and P_2. Since the algorithm of [31, 32] needs to run at most $O(\log M)$ times, the m-th smallest distance in R can be computed in $O(n \log n \log M) = O(n \log^2 n)$ time. The pseudocode of the overall procedure described above is given in [22].

Overall complexity. The $O(\log n)$ iterations of the procedure Cover(P, r) take $O(n \log^2 n)$ time overall, and the $O(\log n)$ queries for the m-th smallest distance overall take $O(n \log^3 n)$ time. This is the dominating term in the running time.

5 Approximate K-Staircase

In this section we give approximation algorithms for the k-staircase problem. First, we show a general algorithm that runs in quadratic time, and then we give a faster approximation algorithm for fixed d.

5.1 An approximation algorithm

In Section 2 we showed that the k-staircase problem is a special case of the asymmetric k-center problem. Previously, [30] gave an $O(\log^* n)$-approximation for the asymmetric k-center problem, so this immediately implies an $O(\log^* n)$ approximation for Problem 1 for any L_α norm. The proof of the next theorem can be found in [22].

▶ **Theorem 11.** *There is an $O(\log^*(n))$-approximation algorithm for the k-staircase problem with $O(n^2(d + \log^2 n))$ running time.*

One can also parameterize on the value of k, in which case [5] provided an $O(\log^* k)$-approximation. Their algorithm requires solving $O(\log n)$ linear programs, so while their algorithm runs in polynomial time, the precise asymptotic time was not stated.

▶ **Theorem 12.** *There exists an $O(\log^*(k))$-approximation algorithm for the k-staircase problem that runs in polynomial time.*

The $O(\log^* n)$-approximation algorithm of [30] can be implemented by performing a binary search over all the $\Theta(n^2)$ inter-point distances. For a value r, deciding if $r < r^*$ or $r \geq r^*$ requires solving a set cover problem. (This is similar to what we do for the exact algorithm in $2D$ in Section 4, except that the set cover instance in general is not solved exactly but only approximately.) In general, even constructing the set cover instance can take quadratic time, and thus approximately solving the entire problem in near-linear time seems difficult. We next use several geometric ideas to come up with a faster bi-criteria approximation (approximation on both k and radius) for any L_α norm, if d is fixed.

5.2 A faster approximation algorithm for fixed d

In this subsection we show a faster approximation algorithm for the k-staircase problem if the dimension d is fixed. Because of the space limit, here we only give a brief overview of our approach. In [22] we provide all the details and the proofs of our method. The main result of this section is the following theorem.

▶ **Theorem 13.** *Given a set of n points $P \subset \mathbb{R}^d$, for constant d, and an integer parameter $k \leq n$, a set $Q \subseteq P$ of size at most $2k$ can be computed in $O(kn\,\mathrm{polylog}(n))$ time, such that $\mathsf{d}_{H_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q)) = O(r^*_\alpha \log^* n)$, where $r^*_\alpha = \min_{Q \subseteq P, |Q| \leq k} \mathsf{d}_{H_\alpha}(\mathsf{SH}(P), \mathsf{SH}(Q))$.*

Our approach is to first get a fast approximation algorithm for the L_∞ norm and then argue that the same solution is a good approximation for any L_α. Intuitively, this will work due to the known inequality, $\|p-q\|_\infty \leq \|p-q\|_\alpha \leq d^{1/\alpha} \|p-q\|_\infty$, for any $p, q \in \mathbb{R}^d$ and any $\alpha \geq 1$, i.e., the L_∞ and any L_α norm differ by a factor that depends only on d and α. Next, we only focus on the L_∞ norm and we denote $r^* = \min_{Q \subseteq P, |Q| \leq k} \mathsf{d}_{H_\infty}(\mathsf{SH}(P), \mathsf{SH}(Q))$ (unlike in the other sections where we use r^* as the optimum distance for any L_α norm).

Our algorithm follows the structure of the algorithm in [30] for the asymmetric k center problem, which has two parts[5]: 1) A decider, such that given a distance r it either returns r is less than r^* or it returns a set $Q \subseteq P$ which is an $O(\log^* n)$-approximation. Their algorithm recursively solves many instances of the set cover problem using the greedy approximation algorithm [15]. Notice that a center in the asymmetric k-center problem covers a set of points within distance r so a set cover instance is indeed an intuitive way to model their problem. 2) A search over all $O(n^2)$ pair distances to find the smallest one where the decider returns an $O(\log^* n)$-approximation.

Both 1) and 2) can be made faster because we are focusing on the L_∞ norm for our k-staircase problem. 1) The sets defined by the set cover instances in [30] correspond to half-open boxes in $\mathbb{R}^d$ for the L_∞ norm. Using geometric data structures, like range trees [1], we can thus run the greedy algorithm for the set cover problem without explicitly constructing the set cover instances, i.e., without finding the points in P contained in each half-open box. 2) Instead of computing the quadratic number of all pairwise directed distances, we use the notion of Well-Separated Pair Decomposition (WSPD) [10, 18, 19, 29]. For a set of n points in $\mathbb{R}^d$, for fixed d, a WSPD can approximate all pairwise Euclidean distances with a near-linear number of distances. Unfortunately, our distance function $\vec{\mathsf{d}}_\infty(\cdot,\cdot)$ is not a metric and as such the known WSPD constructions are not useful. The main idea we use is to project the points to all d-axes separately, and construct the WSPD for each of the projected point sets. The key observation allowing us to construct such WSPDs is that for L_∞ the r^* distance is one of the pairwise distances among the projected 1-dimensional points.

[5] The authors in [30] do not present it this way, however it makes our algorithm easier to explain.

We note that our algorithm finds a set Q with at most $2k$ points. In order to guarantee at most k points with an approximation factor of $O(\log^* n)$ in [30] they make use of the notion of so called CCV points. These are defined and extended in the next section to get a heuristic that works very well in practice. Finally, note that our algorithm works for any dimension d. If d is not a constant, however, then we have an $O(d^{1/\alpha} r^*_\alpha \log^* n)$-approximation algorithm that runs in $O(\text{poly}(d)nk \log^{\text{poly}(d)} n)$ time, where $\text{poly}(d)$ is a linear polynomial of d.

6 A practical heuristic

Here we present a fast heuristic for the k-staircase problem, and provide experimental results showing its efficacy on both real and synthetic data. For simplicity, we present the result only for the L_2 norm and in the full version of the paper [22] we describe how to extend it to any L_α. Missing proofs of all lemmas in this section can be found in [22].

We first provide some intuition for our heuristic. As in the previous sections we look for r^* and the set of k centers using binary search. For a query radius r, we need to be able to decide if $r < r^*$, or $r \geq r^*$. The heuristic attempts to test this, by adapting an approach to solve the standard symmetric k-center problem. The algorithm for the symmetric k-center problem is iterative, and in each iteration it picks some arbitrary center p and removes all points in a ball of radius $2r$ around p. If the procedure stops within k iterations then clearly $r^* \leq 2r$. Otherwise $r^* > r$, because if $r^* \leq r$, then one can argue in each iteration this procedure entirely removes at least one optimal cluster which has not been completely covered yet. To see this, let p_i be the center that the algorithm selected in the ith round. Let o_i be the center from the optimal solution covering p_i. (Since p_i still exists, o_i's cluster has not been fully covered yet.) All remaining points in o_i's cluster are within distance r^* from o_i, which is within r^* from p_i, and hence all remaining points are covered by the $2r$ ball around p_i if $r \geq r^*$. If distances are asymmetric, however, this argument breaks since while p_i is within distance r^* from o_i, this does not imply o_i is within r^* from p_i. To deal with asymmetry, [30] defined the notion of *center capturing vertices* (CCV), where a point p is CCV(r) if whenever the distance from another point to p is less than r, then the distance from p to that point is also at most r. That is, at the resolution of the radius r, the distances involving that point look symmetric. If we could always find a CCV(r) point among the remaining ones, then by the directed triangle inequality the argument would still be valid. However, for general asymmetric distance functions such CCV points may not exist, so we define a new relaxed notion of CCV points allowing imbalance in the directed distances.

▶ **Definition 14** (λ-CCV(r))**.** *Given a point set $P \subset \mathbb{R}^d$, and real numbers $r \geq 0, \lambda > 0$, $p \in P$ is a λ-CCV(r) point, if for all $q \in P$ with $\vec{\mathsf{d}}_2(q,p) \leq r$ it is also true that $\vec{\mathsf{d}}_2(p,q) \leq \lambda r$.*

Note that a 1-CCV(r) point is the same as a CCV(r) point in [30]. Also note that if we can always find a λ-CCV(r) point rather than a CCV(r) point as desired above we will end up with a $(1+\lambda)$ rather than 2 approximation. If r is clear from the context, a CCV(r) (resp. λ-CCV(r)) point is denoted as a CCV (resp. λ-CCV) point. Ultimately our new heuristic works by selecting appropriate λ-CCV points. It is not clear whether such points even exist, however, the following lemma confirms that there is a point that is a $\sqrt{d-1}$-CCV(r) point for *any radius* r, and further, such a point can be found easily (in $O(dn)$ time). For any $u \in \mathbb{R}^d$ define its *weight* as $\mathsf{w}(u) = u_1 + u_2 + \ldots + u_d$.

▶ **Lemma 15.** *For any point set $P \subset \mathbb{R}^d$ with n points, the point $p \in P$ with minimum weight $\mathsf{w}(p)$ is a $\sqrt{d-1}$-CCV(r) point for any distance $r \geq 0$.*

Algorithm 2: **MinWeight**(P, k, r).

Input : P, k, r
Output : Either "$r < r^*$" or a set $Q \subseteq P$ with $|Q| \leq k$ such that $\mathsf{d}_{H_2}(\mathsf{SH}(P), \mathsf{SH}(Q)) \leq (1+\sqrt{d-1})r$

```
X ← P, Q ← ∅
while |Q| ≤ k AND X ≠ ∅ do
    p = argmin_{p∈X} w(p)
    Q ← Q ∪ {p}
    X ← X \ {q ∈ X | d⃗_2(p, q) ≤ (1+√(d−1))r}
if |Q| > k then
    Return "r < r*"
else
    Return Q
```

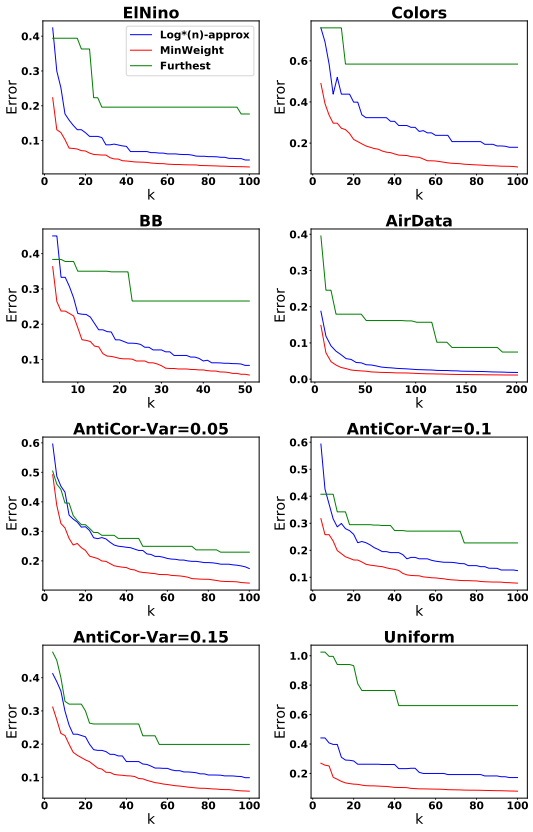

Figure 6.1 Approximation error for L_2 norm.

The decision procedure. To recollect, let r^* be the optimum radius for Problem 1 on a given set P of n points. Our aim is to build a decision procedure which we later use to do a binary search for r^*. Specifically, Algorithm 2 shows a fast procedure **MinWeight**(P, k, r), which either outputs "$r < r^*$" or returns a set Q with $|Q| \leq k$. We were not able to argue that the procedure is always correct when "$r < r^*$" is output, hence the term "heuristic", though the experiments below show that in practice this does not at all seem to be an issue. On the other hand if a set Q is output, we have the following.

▶ **Lemma 16.** *If* ***MinWeight***(P, k, r) *returns a set* $Q \subseteq P$*, then* $|Q| \leq k$ *and* $\mathsf{d}_{H_2}(\mathsf{SH}(P), \mathsf{SH}(Q)) \leq (1+\sqrt{d-1})r$*. The algorithm runs in* $O(dnk)$ *time.*

To see the obstacle in arguing that the algorithm is correct when "$r < r^*$" is output, let Z be an optimal r^* radius solution to Problem 1. We would like to argue that every point $z \in Z$ is at most $\sqrt{d-1}r^*$ from one of the chosen points, as this would imply a $(1+\sqrt{d-1})r^*$ radius covering. However, the issue is that the points we choose in each iteration are $\sqrt{d-1}$-CCV with respect to only the *remaining points* (Lemma 15), for any r. Specifically, consider a point p which is r^*-covered by a point o_i in the optimum solution. It is possible that o_i may cause p to *not* be a $\sqrt{d-1}$-CCV. However, if o_i was removed in some earlier iteration, then later p may become a $\sqrt{d-1}$-CCV point. This could be a problem, as in the worst case it means it is possible that o_i is not $\sqrt{d-1}r^*$-covered by p, and hence all points r^*-covered by o_i are not $(1+\sqrt{d-1})r^*$-covered by p, as we would like to argue.

6.1 Experiments

This section compares the performance of three algorithms: the heuristic MinWeight, the $\log^*(n)$ asymmetric k-center algorithm ([30]) which we call $\text{Log}^*(n)$-approx, and another greedy algorithm Furthest described below. We test on several real and synthetic datasets and use the approximation measure defined in Problem 2 with the asymmetric distance function $\vec{\mathsf{d}}_2(\cdot,\cdot)$. Additionally, in [22], we compare MinWeight and $\text{Log}^*(n)$-approx to the optimal solution (found by brute force) for small synthetic datasets.

All algorithms are implemented in Python on 64-bit machine with 4 3500 MHz cores and 16GB of RAM with Ubuntu 17.10.

Although we do not report running times, the simplicity of MinWeight (Algorithm 2) should be noted. There are k iterations that consist of two steps: greedily choose a point to add to the solution; remove all points within some radius of that point. Not only do both steps take $O(dn)$ time, but they are easily parallelized. Though we don't claim optimal implementations, we mention MinWeight always ran faster than $\text{Log}^*(n)$-approx.

Implementation details.

Log*(n): The authors in [30] simply try all n^2 distance pairs to find the optimal radius r^*. As this is impractical for large datasets, both algorithms (MinWeight and $\text{Log}^*(n)$-approx) instead do a binary search over all distance pairs of the staircase points. Specifically, if the returned set of either algorithm for a given radius is larger than k, we drop the smaller distances; otherwise, we drop the larger distances.

MinWeight: As MinWeight can possibly find a solution of size k on a radius $r < r^*$, the binary search might choose a path where future guesses of r yield no solutions. We store the most recent solution so that if this happens, we can use the stored solution.

Furthest: This is a simple greedy algorithm that iteratively adds the point furthest from the current solution set. That is, given an input set P and a solution set Q_i in round i, it adds $\arg\max_{p\in P}\vec{\mathsf{d}}_2(Q_i,p)$ to Q_i to obtain Q_{i+1} (and repeats until $i=k$). This algorithm does not require searching for an optimal radius r^*.

Datasets. For the experiments we use five of the most common datasets that have been used in other papers related to finding representative points or staircase queries ([3, 6, 8, 13, 28, 34]). More specifically, we use the following datasets.

BB: This is a commonly used ([3, 6, 13]) basketball dataset where each point is a player and the attributes are five statistics: points, rebounds, blocks, assists and fouls. There are 21,961 points in 5-d with 200 points on the staircase.

ElNino: Oceanographic data from the Pacific Ocean, with attributes such as surface temperature, water temperature, and wind speed. There are 178,080 points in 5-d, with 1,183 points on the staircase. This dataset was used in [3, 13] for finding representative points.

AirData: On-time flight data published by the US Department of Transportation, contained in the AirData dataset [6]. Information gathered from 14 carriers flying in January 2015 includes departure delay, taxi in, taxi out, air time, distance, actual elapsed time, and arrival delay. This set has 458,311 points in 7-d, with 6,439 points on the staircase.

Colors: The Colors data set is also commonly used for evaluating staircase and regret sets [28]. The data derives from the HSV color space of a color image, and includes the standard deviation, skewness, and mean of each H, S, and V in the space.

AntiCor: This is a synthetic dataset of anti-correlated points. There are 100,000 points in 5-d that are generated as described in [8]. We test on different values (0.05, 0.1, 0.15) of the variance (Var) that determines the position of the plane. For Var=0.05, there are

7,941 staircase points; for Var=0.1, 1,275 staircase points; and for Var=0.15, 529 staircase points.

Uniform: This synthetic dataset has points sampled uniformly from the unit hypercube which is described in [8]. There are 200,000 points in 7-d with 6,585 points on the staircase.

Results. In each experiment (see Figure 6.1), MinWeight outperforms the other algorithms by at least 50% which suggests that, at least in practice, MinWeight does not erroneously return "$r < r^*$". Indeed, when the input set does not have the structure mentioned after Lemma 16 (where multiple points of an optimal solution are removed in one iteration), MinWeight achieves a $(1+\sqrt{d-1})$-approximation. Even with a $(1+\sqrt{d-1})$-approximation, MinWeight seems to exceed expectations compared against the $\text{Log}^*(n)$-approx. We observed that after the binary search, MinWeight would consistently reach a final guess of r that was 2-4 times smaller than $\text{Log}^*(n)$-approx. One difference between the two algorithms that could explain this is the fact that MinWeight updates its set of $\sqrt{d-1}$-CCV points with respect to the remaining points in each iteration. Specifically, a point which is not $\sqrt{d-1}$-CCV(r) can become $\sqrt{d-1}$-CCV(r) in a later iteration. This indicates that, compared against the 'static' CCV point set in $\text{Log}^*(n)$-approx, there are better points to be chosen conditioned on some set of the previous $\sqrt{d-1}$-CCV(r) points.

6.2 Comparison of algorithms on real data

There are several well known algorithms which attempt to output a set of representative points on the skyline. Comparing the outputs of these algorithms is tricky, as each tries to optimize a different metric. In an attempt at an objective method of comparison, we look at these algorithms and measures on a real data set. Specifically, we look at NBA player statistics over several seasons and use the various staircase algorithms to predict potential all star players in each season (a selection of outstanding players made by the US National Basketball Association). As discussed below, our algorithm consistently selects the most potential all stars, and moreover our error measure appears to most closely track how many potential all stars are selected. Specifically, we compare our Hausdorff measure and our algorithm MinWeight with the measures and the algorithms of the three most cited papers for staircase representation, namely, KolPap [21], k-center [34], and Max-cover [25].

The dataset. For this experiment, we used the Basketball season data set that contains the statistics of each NBA player for each regular season. We use the statistics of each player for five seasons, 2004-2005, ..., 2008-2009. For each player we keep six statistics: total points, number of rebounds, number of assists, number of steals, number of blocks, and the ratio of field goals made over the number of attempted field goals.

The experiment. For each of the five seasons, we run the algorithms MinWeight, KolPap, k-center, and Max-cover and record the $k = 10$ representative players returned, respectively.

The comparison method. Notice that each algorithm returns a set of 10 players S. Each algorithm is attempting to optimize a different measure of error, thus in order to evaluate the quality of the returned set S, we need some neutral method which is different from all the metrics. Towards this end, we define a function $B(S)$ attempting to capture how "good" the set S is, where $B(S) = |T \cap S|$, and T is the set of 50 players selected by NBA experts as

the best in a season that fans could vote to participate in the NBA All Star game.[6] Namely, $B(S)$ returns the number of players in S that were selected as potential players for the NBA All Star game. Such a set of players T always contains the "best" players as selected by NBA experts so we feel it is a fair way to compare the results of the algorithms, so we consider T as the ground truth. Since the notion of skyline approximation is more natural and better suited for users, another possible or even better evaluation would have been by running a real user study.

Assessment of the metric itself. We also obtain some evidence that our metric is somewhat more desirable than the other metrics. Ideally, we would like to have an error function $F(S)$ (to be optimized by an algorithm) such that if $B(S_1) \geq B(S_2)$ then $F(S_1) \leq F(S_2)$, i.e., the better the set is, the smaller the error becomes, or in other words the error has an *inverse relationship* with the quality of the solution.

Format of Table 6.1. To evaluate an algorithm which returns a set S, we show in the table the intersection $B(S) = |T \cap S|$ of S with the ground truth, as well as the error of S for all measures under consideration. So each entry of the table for algorithm X and season Y presents data in the format $B(S)[a, b, c, d]$ where S is the set returned by X and a is the error of S in the Hausdorff distance (our measure), b is the error as per the metric of KolPap, c is the error as per the metric of k-center, and d is the error as per the metric of Max-Cover (the number of uncovered items in P by S).

Table 6.1 Statistics and errors by using different measures and algorithms for $k = 10$.

	2004-2005	2005-2006	2006-2007	2007-2008	2008-2009
MinWeight	8[0.1, 0.25, 0.9, 15]	9[0.18, 0.52, 0.76, 32]	8[0.17, 0.25, 0.79, 31]	8[0.12, 0.67, 0.72, 16]	8[0.09, 0.33, 0.63, 22]
KolPap	7[0.16, 0.18, 0.88, 17]	7[0.22, 0.33, 0.62, 16]	7[0.17, 0.25, 0.79, 30]	6[0.3, 0.34, 0.74, 66]	8[0.14, 0.23, 0.8, 9]
k-center	7[0.34, 0.51, 0.48, 27]	5[0.31, 1.2, 0.50, 57]	5[0.43, 0.75, 0.55, 68]	6[0.34, 0.93, 0.51, 19]	6[0.22, 0.71, 0.49, 49]
Max-cover	7[0.32, 0.44, 0.77, 2]	5[0.39, 0.69, 0.69, 4]	6[0.44, 0.75, 0.67, 1]	6[0.38, 0.59, 0.52, 1]	5[0.37, 0.48, 0.66, 3]

Analysis of the results. From Table 6.1 we can derive the following conclusions: (I) The MinWeight algorithm always returns better results (based on the function B), and (II) the Hausdorff error has an inverse relationship with the quality of the result $B(S)$. We obtain several data points to test this relationship. For a fixed season, the various algorithms return some set S.We plot the points $(B(S), \text{error})$ for each algorithm using the set S it returns and for the errors under the different metrics. If we look at the errors for a fixed measure, the Hausdorff distance error measure satisfies the inverse relationship in almost all the cases. The rest of the measures defined by the previous works do not seem to have this desirable inverse relationship. In the full version of the paper [22] we show the results for $k = 15$ and we provide more figures showing the relationship of the quality $B(S)$ with the error measures.

7 Conclusions

Our work suggests several directions for future research. As shown, the k-staircase problem is an instance of the asymmetric k-center problem and this implies an $O(\log^* n)$-approximation. However, it is an open question whether we can exploit the geometric properties of the

[6] We use `https://www.basketball-reference.com/allstar/` to get the players selected by the experts.

problem to improve the approximation. In particular, can we get an $f(d)$-approximation for the k-staircase problem, where $f(d)$ is a function only depending on the dimension? The practical success of our heuristic and the properties discussed in Section 6 gives some indication this may be possible. Even if this is not possible, one could consider whether relaxing the constraint on having exactly k points to $O(k)$ points helps, i.e., whether a bi-criteria approximation is possible. Alternatively, one can consider if there is an $O(\log^* k)$-approximation for the k-staircase problem that does not require solving multiple LP's [5].

References

1 Pankaj K. Agarwal. Range Searching. In *Handbook of Discrete and Computational Geometry, Second Edition.*, pages 809–837. CRC, 2004. doi:10.1201/9781420035315.ch36.

2 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

3 Pankaj K. Agarwal, Nirman Kumar, Stavros Sintos, and Subhash Suri. Efficient Algorithms for k-Regret Minimizing Sets. In *16th International Symposium on Experimental Algorithms, SEA 2017, June 21-23, 2017, London, UK*, pages 7:1–7:23, 2017. doi:10.4230/LIPIcs.SEA.2017.7.

4 Mugurel Ionut Andreica, Eliana-Dina Tirsa, Cristina Teodora Andreica, Romulus Andreica, and Mihai Aristotel Ungureanu. Optimal geometric partitions, covers and K-centers. *arXiv preprint arXiv:0908.3652*, 2009.

5 Aaron Archer. Two O ($\log^*$ k)-Approximation Algorithms for the Asymmetric k-Center Problem. In *Integer Programming and Combinatorial Optimization, 8th International IPCO Conference, Utrecht, The Netherlands, June 13-15, 2001, Proceedings*, pages 1–14, 2001. doi:10.1007/3-540-45535-3_1.

6 Abolfazl Asudeh, Azade Nazi, Nan Zhang, and Gautam Das. Efficient Computation of Regret-ratio Minimizing Set: A Compact Maxima Representative. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 821–834, 2017. doi:10.1145/3035918.3035932.

7 Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse Approximation via Generating Point Sets. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 548–557, 2016. doi:10.1137/1.9781611974331.ch40.

8 Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The Skyline Operator. In *Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001, Heidelberg, Germany*, pages 421–430, 2001. doi:10.1109/ICDE.2001.914855.

9 Greg Van Buskirk, Benjamin Raichel, and Nicholas Ruozzi. Sparse Approximate Conic Hulls. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2531–2541, 2017. URL: http://papers.nips.cc/paper/6847-sparse-approximate-conic-hulls.

10 Paul B. Callahan and S. Rao Kosaraju. A Decomposition of Multidimensional Point Sets with Applications to k-Nearest-Neighbors and n-Body Potential Fields. *J. ACM*, 42(1):67–90, 1995. doi:10.1145/200836.200853.

11 Wei Cao, Jian Li, Haitao Wang, Kangning Wang, Ruosong Wang, Raymond Chi-Wing Wong, and Wei Zhan. k-Regret Minimizing Set: Efficient Algorithms and Hardness. In *20th International Conference on Database Theory, ICDT 2017, March 21-24, 2017, Venice, Italy*, pages 11:1–11:19, 2017. doi:10.4230/LIPIcs.ICDT.2017.11.

12 Timothy M. Chan, Kasper Green Larsen, and Mihai Patrascu. Orthogonal range searching on the RAM, revisited. In *Proceedings of the 27th ACM Symposium on Computational Geometry, Paris, France, June 13-15, 2011*, pages 1–10, 2011. doi:10.1145/1998196.1998198.

13 Sean Chester, Alex Thomo, S. Venkatesh, and Sue Whitesides. Computing k-Regret Minimizing Sets. *PVLDB*, 7(5):389–400, 2014. doi:10.14778/2732269.2732275.

14 Julia Chuzhoy, Sudipto Guha, Eran Halperin, Sanjeev Khanna, Guy Kortsarz, Robert Krauthgamer, and Joseph Naor. Asymmetric k-center is $\log^* n$-hard to approximate. *J. ACM*, 52(4):538–551, 2005. doi:10.1145/1082036.1082038.

15 Vasek Chvátal. A Greedy Heuristic for the Set-Covering Problem. *Math. Oper. Res.*, 4(3):233–235, 1979. doi:10.1287/moor.4.3.233.

16 Tomás Feder and Daniel H. Greene. Optimal Algorithms for Approximate Clustering. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 434–444, 1988. doi:10.1145/62212.62255.

17 Harold N. Gabow, Jon Louis Bentley, and Robert Endre Tarjan. Scaling and Related Techniques for Geometry Problems. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 135–143, 1984. doi:10.1145/800057.808675.

18 Sariel Har-Peled. *Geometric approximation algorithms*, volume 173. American mathematical society Boston, 2011.

19 Sariel Har-Peled and Manor Mendel. Fast Construction of Nets in Low-Dimensional Metrics and Their Applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006. doi:10.1137/S0097539704446281.

20 Refael Hassin and Arie Tamir. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10(7):395–402, 1991.

21 Vladlen Koltun and Christos H. Papadimitriou. Approximately Dominating Representatives. In *Database Theory - ICDT 2005, 10th International Conference, Edinburgh, UK, January 5-7, 2005, Proceedings*, pages 204–214, 2005. doi:10.1007/978-3-540-30570-5_14.

22 Nirman Kumar, Benjamin Raichel, Stavros Sintos, and Gregory Van Buskirk. Approximating Distance Measures for the Skyline. http://utdallas.edu/~benjamin.raichel/stair.pdf.

23 Nirman Kumar and Stavros Sintos. Faster Approximation Algorithm for the k-Regret Minimizing Set and Related Problems. In *Proceedings of the Twentieth Workshop on Algorithm Engineering and Experiments, ALENEX 2018, New Orleans, LA, USA, January 7-8, 2018.*, pages 62–74, 2018. doi:10.1137/1.9781611975055.6.

24 H. T. Kung, Fabrizio Luccio, and Franco P. Preparata. On Finding the Maxima of a Set of Vectors. *J. ACM*, 22(4):469–476, 1975. doi:10.1145/321906.321910.

25 Xuemin Lin, Yidong Yuan, Qing Zhang, and Ying Zhang. Selecting Stars: The k Most Representative Skyline Operator. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 86–95, 2007. doi:10.1109/ICDE.2007.367854.

26 Matteo Magnani, Ira Assent, and Michael L. Mortensen. Taking the Big Picture: representative skylines based on significance and diversity. *VLDB J.*, 23(5):795–815, 2014. doi:10.1007/s00778-014-0352-3.

27 S. Mentzer. Approximability of Metric Clustering Problems. preprint on researchgate.net, 2016. URL: https://www.researchgate.net/publication/242489373_Approximability_of_Metric_Clustering_Problems.

28 Danupon Nanongkai, Atish Das Sarma, Ashwin Lall, Richard J. Lipton, and Jun (Jim) Xu. Regret-Minimizing Representative Databases. *PVLDB*, 3(1):1114–1124, 2010. doi:10.14778/1920841.1920980.

29 Giri Narasimhan and Michiel H. M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007.

30 Rina Panigrahy and Sundar Vishwanathan. An O ($\log^* n$) Approximation Algorithm for the Asymmetric p-Center Problem. *J. Algo.*, 27(2):259–268, 1998. doi:10.1006/jagm.1997.0921.

31 J. Salowe. *Selection Problems in Computational Geometry*. PhD thesis, Rutgers University, 1987.

32 Jeffrey S. Salowe. L-Infinity Interdistance Selection by Parametric Search. *Inf. Process. Lett.*, 30(1):9–14, 1989. doi:10.1016/0020-0190(89)90166-X.

33 Malene Søholm, Sean Chester, and Ira Assent. Maximum Coverage Representative Skyline. In *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016.*, pages 702–703, 2016. doi:10.5441/002/edbt.2016.95.

34 Yufei Tao, Ling Ding, Xuemin Lin, and Jian Pei. Distance-Based Representative Skyline. In *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, pages 892–903, 2009. doi:10.1109/ICDE.2009.84.

35 Stan PM Van Hoesel and Albert PM Wagelmans. On the p-coverage problem on the real line. *Statistica Neerlandica*, 61(1):16–34, 2007.

36 Rui Xu and Donald C. Wunsch II. Survey of clustering algorithms. *IEEE Trans. Neural Networks*, 16(3):645–678, 2005. doi:10.1109/TNN.2005.845141.